Midterm Exam

CMSY-217, Spring 2013

Section 1. Answer True or False to each of the following statements:

_____ 1. A recursive Java method cannot be called by itself, be called directly, nor be called indirectly through another method.

_____ 2. The `binarySearch` method in the `Arrays` class has a runtime efficiency of $O(\log n)$.

_____ 3. The `List` interface is a subinterface of the `Collections` interface.

_____ 4. The process by which the Java complier replaces generic type parameters and arguments within a class or method is called deletion.

_____ 5. The `Stack` class represents a last-in-first-out (LIFO) stack of objects.

_____ 6. The x-coordinate in Java is the horizontal distance moving right from the left side. The y-coordinate in Java is the vertical distance moving down from the top edge.

_____ 7. All of the type-wrapper classes in Java are subclasses of the `Number` class.

_____ 8. Graphical applications are typically implemented in Java by first creating a `JPanel` and then calling its add method with a `JFrame` as the argument.

Section 2. Circle the letter of the best answer for each question:

9. In the JVM, the current method executing is always the method whose activation record is:
   (a) at the bottom of the runtime stack.
   (b) at the top of the runtime stack
   (c) never placed on the runtime stack.
   (d) second from the top of the runtime stack, just below the record for the previous method call.
10. What is the runtime efficiency of the Merge Sort algorithm presented in the textbook?
   (a) $O(1)$
   (b) $O(n)$
   (c) $O(n \log n)$
   (d) $O(n^2)$

11. Which interface requires the `compareTo` method to be implemented?
    (a) `Comparable`
    (b) `Enumerable`
    (c) `Orderable`
    (d) `Tractable`
12. If the upper bound of a generic class is an interface, the Java keyword used to specify it is:
    (a) `bounds`
    (b) `extends`
    (c) `implements`
    (d) `limits`
13. The `LinkedList` class can be used to implement which of the following custom data structures?
    (a) stack
    (b) queue
    (c) double-ended queue
    (d) all of the above
14. Which of the following is equivalent to the constant `Color.GREEN`?
    (a) `new Color(255,0,0);`
    (b) `new Color(0,255,0);`
    (c) `new Color(0,0,255);`
    (d) `new Color(255,255,0);`
15. A benefit of generic collections is
    (a) compile-time checking
    (b) an explicit cast is not required when removing items
    (c) runtime safety
    (d) all of the above
16. What interface is designed to provide an alternative to the natural ordering of a collection?
    (a) `Comparable`
    (b) `Comparator`
    (c) `OrderAlernator`
    (d) `Quantifiable`

Section 3. Answer the following questions:

17. The factorial function can be defined recursively as:

$$n! = \begin{cases} 1 & \text{if } n = 0 \\ n \times (n-1)! & \text{if } n > 0 \end{cases}$$

Complete the recursive `factorial` method using this definition.

```
public static int factorial(int n)
{
   if (          )
   {




   }
   else
   {




   }
}
```

18. The selection sort algorithm makes $n - 1$ passes through an array of size n. On the first pass, it begins with the first element of the array and examines each successive element to determine the index of the smallest element; the first element and the smallest element are then swapped. On the second pass, it begins with the second element of the array and examines each successive element to determine the index of the smallest remaining element; the second element and the smallest remaining element are then swapped. The process repeats until the $n-1$ pass when the final two elements are examined and (possibly) swapped. Complete the `selectionSort` method below which uses the selection sort algorithm to sort an array of integers.

```
public static void selectionSort(int a[])
{
   int i, j, smallestIndex;

   for (                                  )
   {
      smallestIndex = i;
      for (                                  )
      {
         if (                          ) smallestIndex = j;
      }
      swap(a,i,smallestIndex);
   }
}
```

```
private static void swap(int a[], int i, int j)
{
   int temp = a[i];
   a[i] = a[j];
   a[j] = temp;
}
```

19. Provided that an instance of the `Stack<Character>` class called `stack` is in scope, write a method called `printReverse` that takes a `String` argument and uses `stack` to print its characters in reverse.

```
public static void printReverse(String s)
{
   for (                              )
   {



   }

   while(                          )
   {



   }

   System.out.println();
}
```

20. Write a self-referential, generic class called `Node` that has two `private` members - one which can store data of the parameterized type and another which is a link to the next node in the list.

21. You have implemented a custom generic list class that is doubly-linked. The references to the first and last nodes in the list are generic type parameters named `first` and `last`, respectively. Write a method called clear which removes all elements from the list.

```
public void clear()
{









}
```

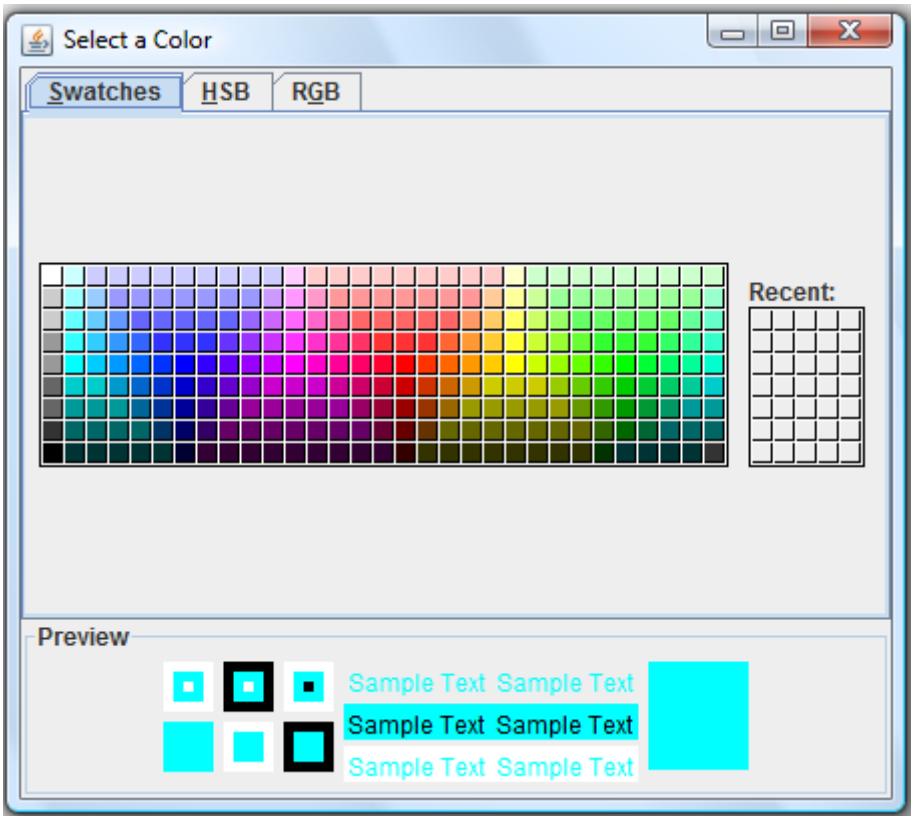
22. Complete the `paint` method below so that running the program displays the message shown. Note that the font color is white, and the text is positioned at the coordinates $(40, 65)$.


```
public void paintComponent(Graphics g)
{
  super.paintComponent(g);
  setBackground(Color.RED);
  setFont(new Font("Courier", Font.BOLD, 24));









}
```

23. Rewrite the following code using generic collections. Be sure to remove any unnecessary code.

```
ArrayList list = new ArrayList();
list.add("someString");
String s = (String) list.get(0);
```

24. Complete the `main` method below so that running the program displays the following dialog box




```
public static void main(String args[])
{
   JFrame frame = new JFrame("Select a Color");




   frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
   frame.setSize(450, 400);
   frame.setVisible(true);
}
```

Section 4. Circle the letter of the best answer for each question:

25. Given the following method, what is the the return value for the method call `compute(7)`?

```
public static int compute(int n)
{
   if (n == 0)
      return 0;
   else if (n == 1)
      return 1;
   else
      return compute(n-1) + compute(n-2);
}
```

(a) 8

(b) 13

(c) 21

(d) 34

26. Suppose the array `[40, 17, 45, 82, 62, 32, 30, 44, 93, 10]` is passed to the method below to perform an insertion sort. What are the contents of the array after the first iteration of the `for` loop?

```
public static void insertionSort(int[] data)
{
   int insert;

   for (int next = 1; next < data.length; next++)
   {
      insert = data[next];

      int moveItem = next;

      while (moveItem > 0 && data[moveItem - 1] > insert)
      {
         data[moveItem] = data[moveItem - 1];
         moveItem--;
      }

      data[moveItem] = insert;
   }
}
```

(a) `[10, 17, 45, 82, 62, 32, 30, 44, 93, 40]`

(b) `[17, 40, 45, 82, 62, 32, 30, 44, 93, 10]`

(c) `[17, 40, 45, 62, 32, 30, 44, 82, 10, 93]`

(d) `[10, 17, 30, 32, 40, 44, 45, 62, 82, 10]`

27. Consider the following code segment.

```
ArrayList<String> list = new ArrayList<String>();

list.add("P");
list.add("Q");
list.add("R");
list.set(2, "s");
list.add(2, "T");
list.add("u");
System.out.println(list);
```

What is printed as a result of executing the code segment?

(a) `[P, Q, R, s, T]`

(b) `[P, Q, s, T, u]`

(c) `[P, Q, T, s, u]`

(d) `[P, T, Q, s, u]`

(e) `[P, T, s, R, u]`

28. What is the output if you compile and execute the following Java application?

```
import java.util.*;

public class BarSort
{
   public static void main(String args[])
   {
      Number a[] = {3, 1, 4, 6, 5, 2, 0};
      barSort(a);
      System.out.println(Arrays.toString(a));
   }

   public static <T> void barSort(T[] a)
   {
      Arrays.sort(a);
   }

   public static void barSort(Integer[] a)
   {
      Arrays.sort(a, Collections.reverseOrder());
   }
}
```

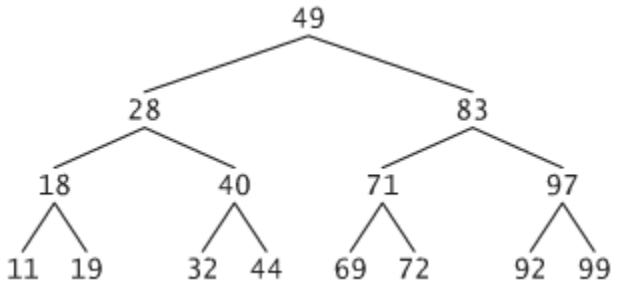
(a) `[Ljava.lang.Integer;@3e25a5`

(b) `[0, 1, 2, 3, 4, 5, 6]`

(c) `[6, 5, 4, 3, 2, 1, 0]`

(d) Compilation fails.

29. A binary search tree is constructed using a generic self-referential tree node class. Fifteen nodes are added and the tree can be represented visually by the following figure



If the following recursive method is called on the root node of the tree, what is the output?

```
public static <E> void traverse(TreeNode<E> node)
{
   if (node.getLeft() != null) traverse(node.getLeft());
   if (node.getRight() != null) traverse(node.getRight());
   System.out.print(node.getData() + " ");
}
```

(a) `49 28 18 11 19 40 32 44 83 71 69 72 97 92 99`

(b) `11 18 19 28 32 40 44 49 69 71 72 83 92 97 99`

(c) `11 19 18 32 44 40 28 69 72 71 92 99 97 83 49`

(d) `49 28 83 18 40 71 97 11 19 32 44 69 72 92 99`

30. What is the output when the following `main` method is compiled and run?

```
public static void main(String args[])
{
   Integer a[] = {3, 1, 4, 1, 5, 9, 2};
   List<Integer> list = new LinkedList<Integer>(Arrays.asList(a));
   list.remove(2);
   System.out.println(list);
}
```

(a) `[3, 1, 4, 1, 5, 9]`

(b) `[3, 1, 1, 5, 9, 2]`

(c) Compilation fails.

(d) An exception is thrown at runtime.

31. Which of the following is *not* a subinterface of the `Collection` interface?

    (a) `List`

    (b) `Map`

    (c) `Queue`

    (d) `Set`

32. Which of the following classes does not implement a generic `Comparable` interface?

    (a) `Arrays`

    (b) `Character`

    (c) `Integer`

    (d) `String`

Section 5. Fill in the blanks in each of the following statements:

33. When implementing a Java program program to compute factorials, the precision of the `int` and `long` types was exceeded at 12! and 21!, respectively. In order to compute larger factorials, one may use the _____ or _____ class from the `java.math` package.

34. Two iterative sorting algorithms that we studied and implemented in Java were the _____ and _____ .

35. Although the classes of the Java Collections Framework can only be used to store reference type objects, Java can automatically convert primitive-values to type-wrapper objects using _____ and convert from type-wrapper objects to primitive-type values using _____.

36. _____ and _____ enable you to specify a set of related methods or classes with a single declaration.

37. Since the stack data structure is a constrained version of a list, stacks may be implemented using classes that implement the `List<T>` interface such as the _____ or _____.

38. The `Graphics` API provides a method to display text called_____ and a method to display solid boxes called _____.

39. For the `QuickSort` class that you implemented, the two static methods were named _____ and _____ .

40. When writing an application that uses the `Graphics` API, you often need to import classes from the
    java._____ and the javax._____ packages.